NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DETECTING AND MEASURING TEMPORAL PHENOMENON WITH HIGH RESOLUTION SATELLITE IMAGERY

by

Andrew J. Hittner

September 2003

Thesis Advisor: Richard Olsen
Co-Advisor: Richard Harkins

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2003	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Detecting and Measuring Temporal Phenomenon with High Resolution Satellite Imagery		**5. FUNDING NUMBERS**	
6. AUTHOR Andrew J. Hittner			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE** A	

13. ABSTRACT *(maximum 200 words)*

A new operating mode for imaging satellites has been developed. The non-imaging mode is designed to provide high temporal resolution data time varying targets. The idea is to use linear pushbroom arrays with kilohertz sampling rates to not only detect, but also measure the frequency of temporally varying targets. For satellite operations this will involve satellite slewing at a rate which cancels the effects of satellite orbital motion. This concept was explored here with a laboratory simulation of an intended test target consisting of windmills at the Ponnequin Wind Farm in Colorado. Test images were acquired and processed using an inexpensive camera and MATLAB. Results indicate the approach is viable and should produce distinct and useful signatures in time and frequency domain analysis.

14. SUBJECT TERMS Detecting motion with imagery satellites			15. NUMBER OF PAGES 59
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

DETECTING AND MEASURING TEMPORAL PHENOMENON WITH HIGH RESOLUTION SATELLITE IMAGERY

Andrew J. Hittner
Captain, United States Army
B.S., United States Military Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Andrew J. Hittner

Approved by: Richard Olsen
                 Thesis Advisor

                 Richard Harkins
                 Co-Advisor

                 Rudolf Panholzer
                 Chairman, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A new operating mode for imaging satellites has been developed. The non-imaging mode is designed to provide high temporal resolution data time varying targets. The idea is to use linear pushbroom arrays with kilohertz sampling rates to not only detect, but also measure the frequency of temporally varying targets. For satellite operations this will involve satellite slewing at a rate which cancels the effects of satellite orbital motion. This concept was explored here with a laboratory simulation of an intended test target consisting of windmills at the Ponnequin Wind Farm in Colorado. Test images were acquired and processed using an inexpensive camera and MATLAB. Results indicate the approach is viable and should produce distinct and useful signatures in time and frequency domain analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I.   INTRODUCTION

A.   PURPOSE OF RESEARCH

The end of the Cold War in the late 1980s changed the mission of America's intelligence collecting agencies.   There is no longer a single Soviet threat.   Instead there are countless small, loosely organized terrorist organizations.   These terrorists are not capable of directly attacking the democracies of the world so they attempt to disrupt the free world's economic systems and undermine their governments through violent acts such as the suicide attacks on September 11, 2001.   These organizations have no home country so they live in caves and underground bunkers.   They produce chemical and biological weapons in factories disguised as legitimate businesses, and they operate from inside countries whose governments are not friendly to the democracies of the world.

The detection of facilities being used for weapons production, and the diagnosis of such facilities capabilities is a difficult problem.   Traditional remote sensing modalities, including high spatial resolution imagery, may not address such problems.

Targets of military or strategic military interest may, however, have detectible signatures associated with human or machine activity.   One example is the rotation or vibration of roof-top air ventilators.   Activity within a facility will be reflected in the motion of such devices.   If the rotational velocity can be inferred, or similar information obtained, quantities such as the volume of air being moved can be estimated.

The purpose of the research effort described here is to develop a technique for observing temporal signatures (vibrations) using high spatial resolution (satellite) imaging systems.   The approach will be to take advantage of the high sampling rate of linear, push-broom imagers.

B.   OBJECTIVE

The ultimate objective of this project is to observe the motion or velocity of a ground target using a satellite imagery system.   This will be accomplished by forcing the satellite to stare at a strip of land that contains the rotating (or vibrating) object for several seconds, and then analyzing the data row by row in both the time and frequency domains

to determine the rotational velocity. Our initial satellite tests will involve large obvious targets – windmills used for power generation

The objective of this thesis is to use a laboratory model to demonstrate the data manipulation and mathematical techniques needed to process the satellite data. This model was designed to resemble the power generation windmills at Ponnequin wind farm in northern Colorado which are the targets for the satellite data (see Chapter V for details on Ponnequin wind farm). A follow on student will attempt to implement the techniques of this thesis with actual satellite data to complete the overall project. The laboratory target is a model windmill standing 12 inches high. The windmill has three blades and is powered by a 3 volt motor which gives it a rotational velocity of approximately 22 rotations per minute.
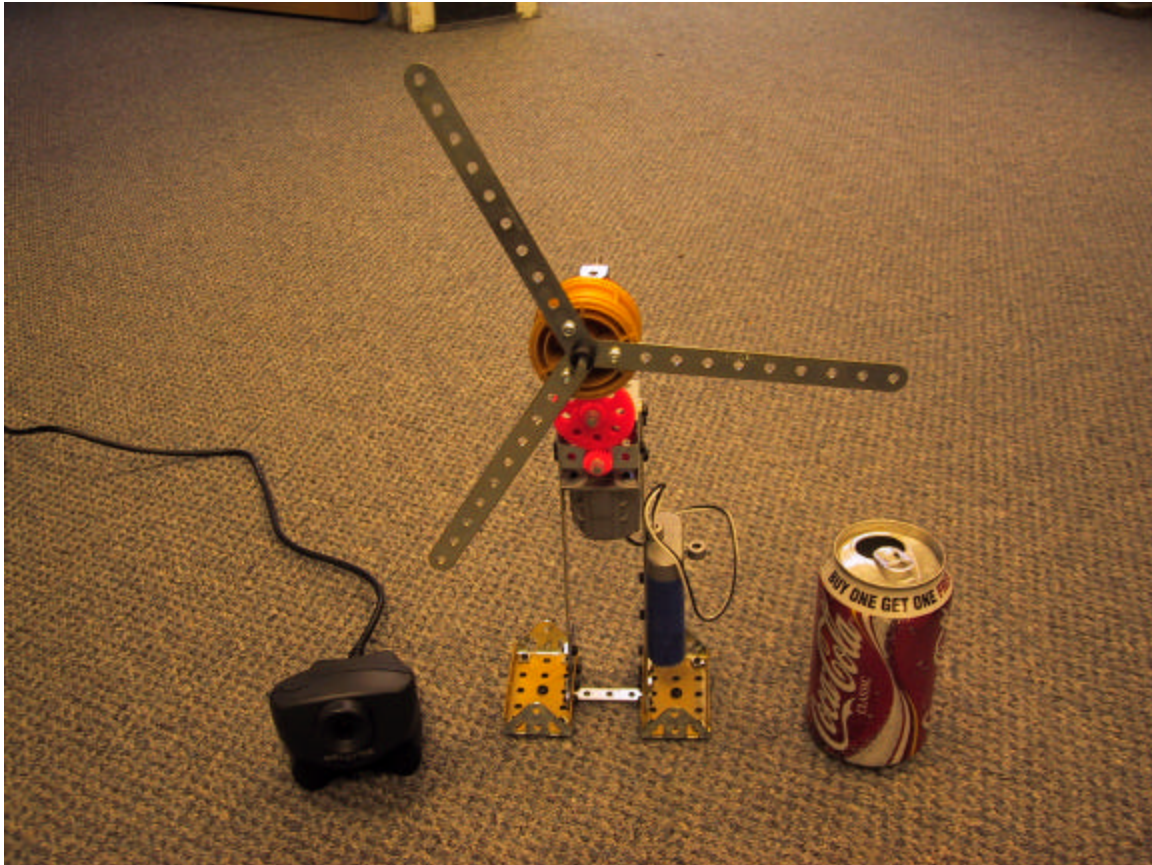

Figure 1 – Model windmill and webcam

II.   SOLID STATE ARRAYS

A.   INTRODUCTION

Electro-optical imaging requires the use of some type of solid state array.   In this thesis the solid state array is a webcam with a rectangular array.   The follow on work to this thesis will utilize a linear solid state array on a commercial imaging satellite.   When photons of light strike a solid state array they build up an electric charge which is eventually converted into a digital picture inside a computer.   There are three basic functions of a solid state array: (1) charge collection, (2) charge transfer, (3) the conversion of charge into a measurable voltage[1].   These measurable voltages are then read into a computer.   A bright light consists of many photons which will create a high charge on the solid state array.   This high charge creates a large measurable voltage which is read into the computer as a large number.   Image interpretation software displays pixels with a large number brighter than pixels with low numbers.

B.   PHOTODETECTION

The photodetection process is rather simple.   A photon of light strikes a pixel on the array where it is absorbed.   This dislodges an electron.   As additional photons strike the pixel, more electrons are dislodged and an electric charge is generated in the pixel[2].   Of course, not every photon that reaches the lens of the camera will dislodge an electron in a pixel of the solid state array.   Some photons will be reflected by the material on the surface of the array or by the lens of the camera; however, if care is used in manufacturing the lens and solid state array, then reflectance will be constant across all pixels, and image processing software can correct for variations in intensity[3].

If the camera is imaging an area with a very bright light then a pixel of the array may absorb too many photons of light.   This results in more dislodged electrons than the pixel's well can hold so some electrons will spill out into other pixels.   This process is called blooming[4].   One technique for preventing blooming is reducing the amount of time the solid state array is exposed to light (think of this as closing the shutter on a camera more quickly).   Of course there are other, more elegant techniques for handling the

problem which are beyond the scope of this thesis. Blooming was prevented in the laboratory experiments by limiting the intensity of light in the room when the windmill was imaged.

C.   ARRAY OPERATION

A solid state array is a series of charge coupled registers[5]. For simplicity, the registers in this example consist of two gates. Many solid state arrays use three or four gates to improve efficiency of operation, but two gates are sufficient to explain the process. When a voltage is applied to a gate a potential is created which creates a well below the gate where electrons are stored (see figure 2). Initially, a voltage is applied to gate 1 which creates a well below it. While this voltage is active, the solid state array is exposed to the light reflected from the area being imaged. Photons strike the pixel dislodging electrons which fall into well 1. This generally lasts a millisecond or two depending on how bright the area being imaged is, and it completes the charge collection phase. Now the array must transfer the charge of each individual pixel to a device (a capacitor) which is capable of converting the charge into a voltage potential. First a voltage potential is applied to gate two which causes well two to open. The charge (electrons) quickly flows into well two until the charge is evenly balanced between wells one and two. Next the voltage potential on well one is removed causing the remaining charge in well one to flow into well two[6].
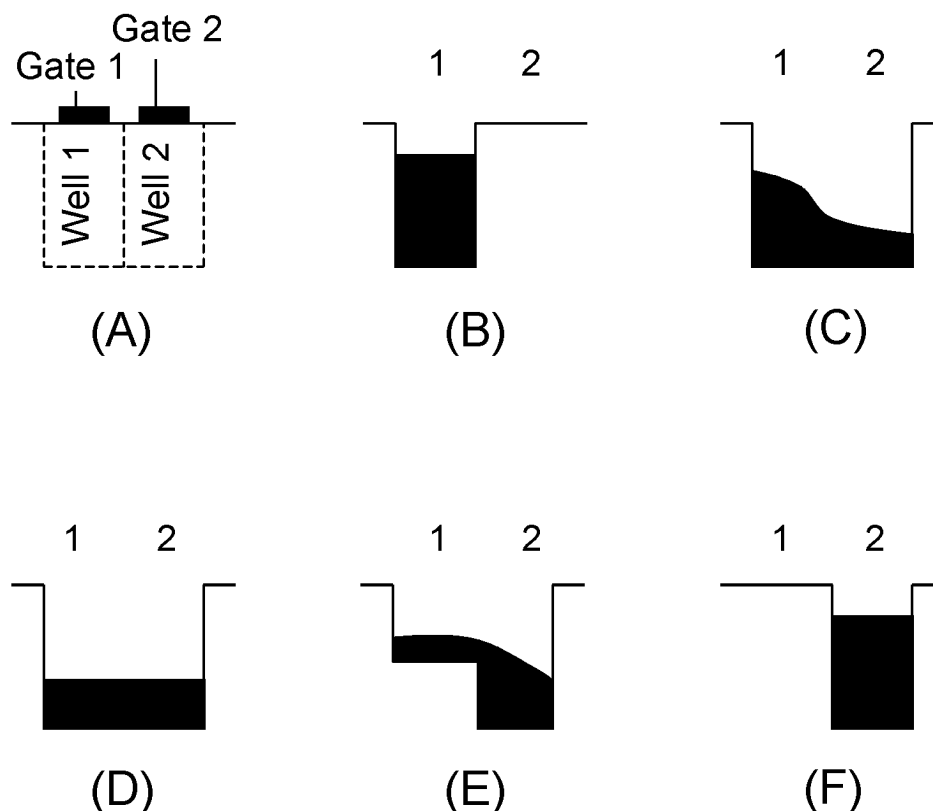
Figure 2 – This diagram depicts the transfer of charge from one well to another inside a single pixel of a solid state array. (A) This shows both wells. (B) A voltage is applied to gate 1, but not to gate 2. The pixel is exposed to light and a charge is collected in well 1. (C) A charge is applied to gate 2 and the charge begins to flow into well 2. (D) The charge balances between wells 1 and 2. (E) The voltage on gate 1 is reduced causing charge to flow into well 2. (F) All voltage on gate 1 is removed causing all charge to flow to well 2.

This demonstrates how charge is transferred inside a single pixel. However, the charge must eventually reach a device which can convert it to a measurable voltage. This process is depicted in figure 3. This figure depicts the array as a series of columns with a horizontal readout register. Charge passes down the columns in the manner described above until it reaches the bottom. The bottom row transfers the charge to the right where each pixel is converted to a voltage which is measured and stored[7]. Note that even though the pixels travel down the array in columns, they are read out in rows. A computer can easily assemble these rows into an image file or process the data in any

number of different ways. It is important to note that solid state arrays can take many different shapes. They can be square (as depicted below), rectangular (like most digital cameras), or linear (like imagery satellites).
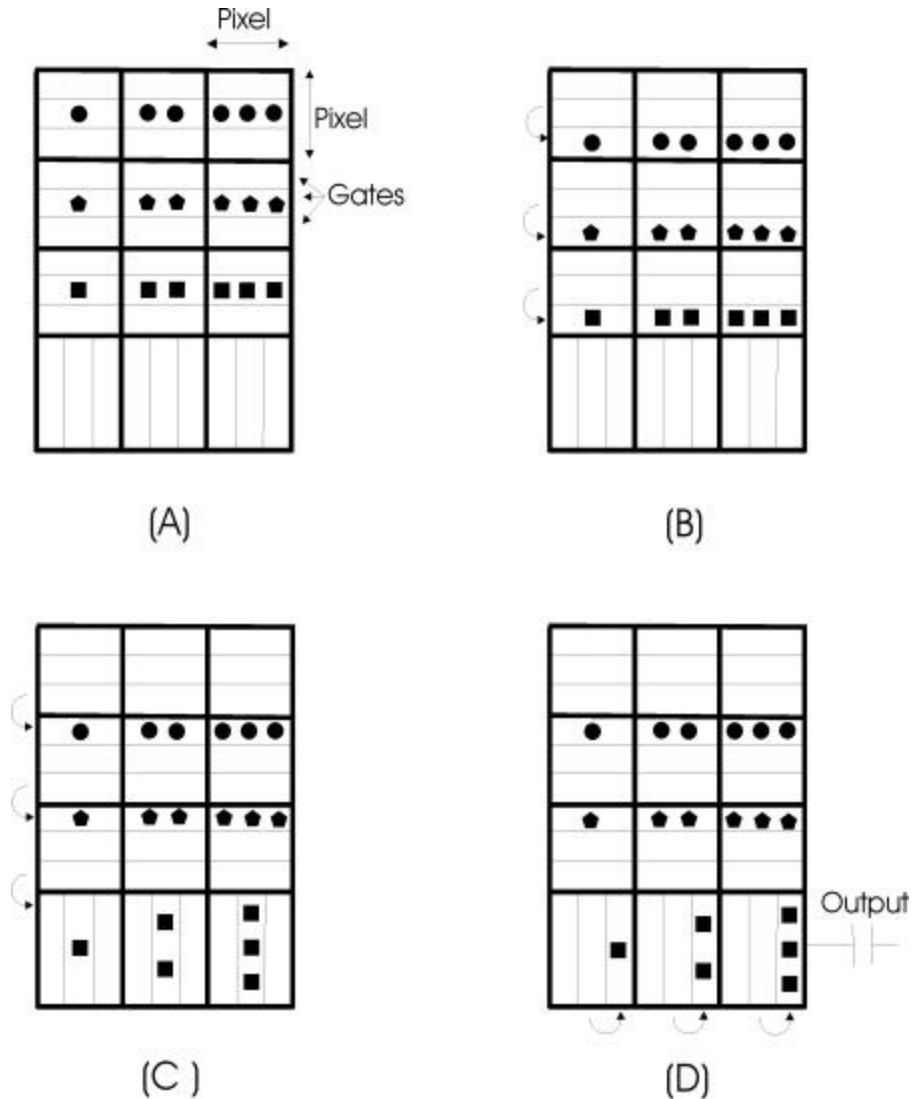
Figure 3 – This figure shows charge transferring down an array of pixels and to an output device. (A) The array is exposed to light and charge collects in each pixel. (B) Charge is transferred down 1 well within each pixel. (C) Charge transfers down again, this time the bottom row enters the horizontal readout register. (D) Charge in the readout register begins moving to the right where it is converted into a measurable voltage and read into computer memory. The entire bottom row must move all the way to the right before another row can enter the readout register.

D.   TIME DELAY AND INTEGRATION

Time delay and integration (TDI) is a technique used to improve the signal to noise ratio of an imaging system where either the camera or the target is moving linearly. Improving the signal to noise ratio results in a higher quality image, and, in some cases, it increases the pixel resolution. The process works by exposing multiple detectors to the target and adding their collected charges[8]. This process is best described with an example.

In this example the solid state array is stationary and the target is moving from top to bottom. The target enters the camera's field of view and the photons reflected off the target strike the first detector creating a charge in the top well (see figure 4). As the target moves down, it leaves the field of view of the first detector and enters the field of view of the second detector. Again, photons reflected off the target strike the detector creating a charge in the second well. While this is happening, the charge from the top well passes down to the second well. This results in the second well having twice as much charge as the first well originally had. Note that at this time there is no charge left in the top well. This process continues until the array runs out of detectors[9].
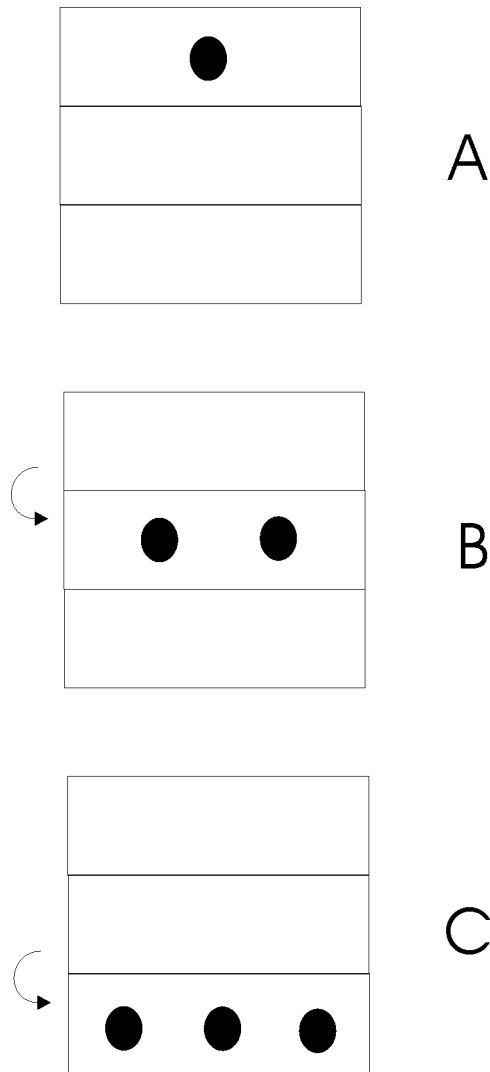
Figure 4 – Time delay and integration. The target in this example is moving from top to bottom. (A) The target enters the field of view of the first detector and a charge is collected. (B) The target moves into the field of view of the second detector and charge is collected. The charge from the first detector is simultaneously transferred to the second detector. (C) The target moves into the field of view of the third detector and a charge is collected. The charge from the second collector is transferred down and added to that collected by the third detector.

Time delay and integration is not a particularly suitable mode for collecting data in this thesis because the rotational motion of the target would be lost in the data. In fact, the blades of the model windmill would appear as a solid disk if TDI was used extensively. The technique is described here because the Quickbird imaging satellite

uses TDI in the panchromatic sensor to improve its sensitivity, as do most similar systems. In the case of the satellite systems, this will lead to blur.

THIS PAGE INTENTIONALLY LEFT BLANK

III.   QUICKBIRD COMMERCIAL IMAGING SATELLITE

A.   INTRODUCTION

The first Quickbird commercial imaging satellite was launched on October 18, 2001 from Vandenberg Air Force Base, California into a polar orbit inclined at 98°. The satellite is capable of achieving 61 centimeter resolution at nadir with its panchromatic sensor and 2.44 meter resolution multispectral. It has a target revisit time of about three days for best resolution, but if the customer is willing to sacrifice resolution by using a greater look angle, then revisit time can be reduced to as little as one day. The satellite orbits at an altitude of 450 km which gives it a period of 93.4 minutes. It has 128 Gbits of onboard storage which it can transmit to ground at 320 Mbps through an X-band downlink[10].

B.   SPECTRAL BANDWIDTHS

There are two sensors on the Quickbird satellite: panchromatic and multispectral. Both sensors use the same optics, but the panchromatic sensor has a spatial resolution four times greater than the multispectral sensor. The pixel pitch is 4 times narrower for the pan detector. As noted above, the intrinsically higher resolution is compromised for our purposes by 10-20 steps of TDI in the along-track direction. The fact that the panchromatic detector uses TDI makes data from this sensor a poor choice for this project.

The second sensor on Quickbird is mulitspectral which measures light intensity in 4 distinct wavelengths: blue, green, red, and near infrared. The actual sensor material used in the multispectral sensor detects light in each of the 4 wavelengths. It separates blue, green, red, and near infrared photons with filters placed over the sensors[11]. The physical layout of the sensors is depicted in figure 6.

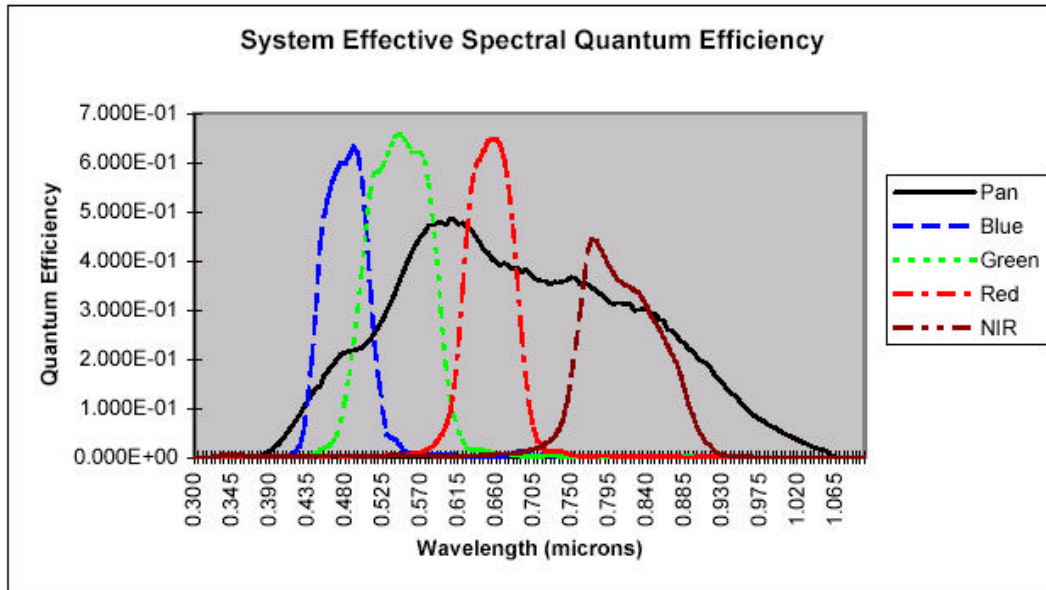**Figure 5 – Quantum efficiency for each spectral band of Quickbird imaging satellite.**
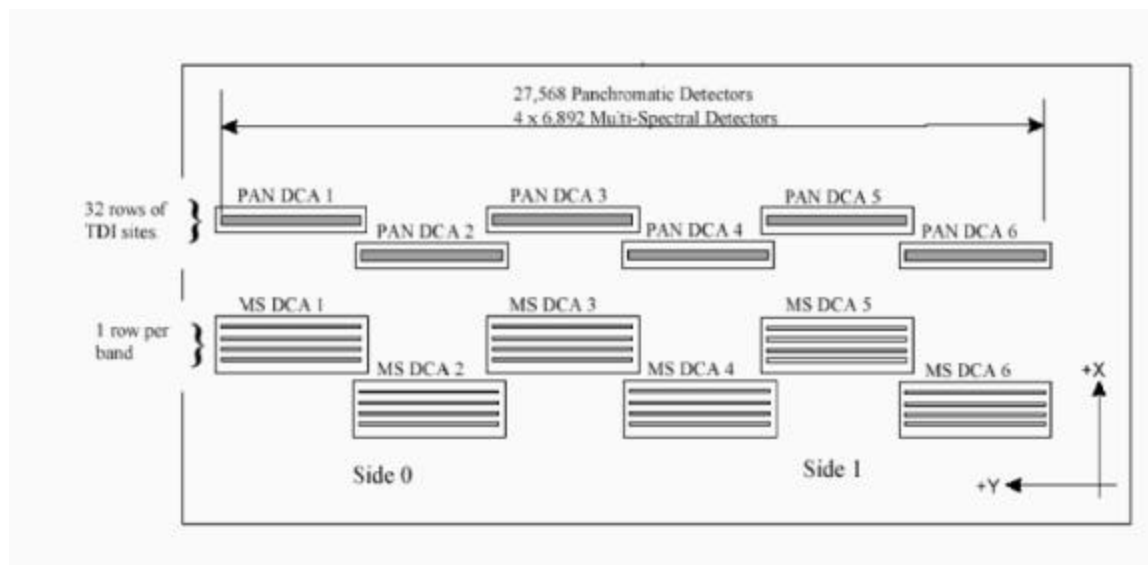

Figure 6 – Quickbird Sensor Array

There are several aspects of the layout that merit discussion. First is that each of the two types of sensor (panchromatic and multispectal) consist of 6 separate subarrays. Each subarray is slightly displaced from the two neighboring subarrays so that their ends

12

overlap. This prevents holes in the image from subarray joints and is compensated for with software from the Kodak company. The software is proprietary so details of how the correction occurs are not available[12]. Note also the multispectral subarrays. Each subarray consists of 4 linear arrays which in turn collect photons consistent with the wavelengths for blue, green, red, and near infrared. Because there are 4 linear arrays collecting light from different wavelengths, TDI is not used. The lack of TDI is related to the lower spatial resolution (16 times larger pixels) in the MSI array. This makes the multispectral sensor a better choice for this project.

C.   MODES OF OPERATION

The primary mode of operation for the Quickbird satellite is commonly called the pushbroom technique. This technique involves exposing the linear array to a strip of ground (16.5 km wide for Quickbird). The width of the strip will vary based on look angle, but for this example assume it is 1 meter. The data from that scan must be stored quickly so the array can be exposed to another 16.5 km strip one meter in front of the previous strip (see figure 7). Quickbird continues scanning strip after strip until the entire area of concern is imaged. Software on the ground then assembles the data into a picture.
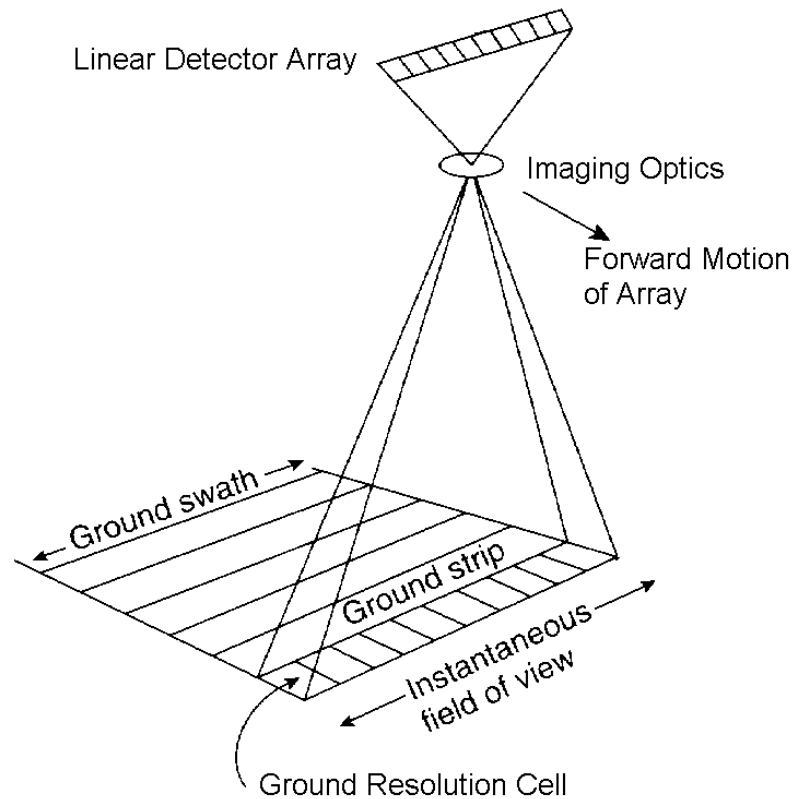
Figure 7 – Pushbroom system for linear solid state array

The Quickbird satellite was designed to operate at a lower altitude than other satellites of its type to provide a higher spatial resolution. Operating at the lower altitude required modification to the satellite operation to include a modest slewing motion as the satellite passes over the target. This slewing allows a longer imaging time without causing motion blur. For the panchromatic sensor, TDI levels of 10-20 are then used to further balance exposure times against motion blur.

It is possible to force the satellite to remain focused on the same strip of land and take hundreds of images. If a moving object is in the FOV, a temporal signature should be observable. Unfortunately, extensive reprogramming of the satellite is needed to force it to remain focused on the same place. In addition to the time and expense involved in writing the code for the new operating mode, there is considerable risk to the satellite owner (in this case Digital Globe) because there is some possibility that the satellite may

not respond properly and might incur some damage. Digital Globe is presently studying this mode for safety.   Fortunately, there is another mode of operation, which some systems are already programmed to operate in, which will also provide the means to conduct this experiment.

Electro-optical imaging systems should be calibrated before use to ensure the image it produces is an accurate representation of the target. Distortions can occur from many different sources such as imperfections in the lens or material coating the solid state array, dark current within the array, or slightly different well sizes within the array which could occur if the voltage applied to each well varies slightly throughout the array. Even if all these factors could be completely eliminated, the satellite itself degrades in ability due to exposure to radiation and debris in space.   Therefore, a calibration mode is included in the operating mode for some systems. To calibrate the sensor arrays, the satellite operators pivot the satellite 90° (refer to figure 6). This means the satellite is now traveling parallel to the scan line (this is the y direction in the figure). The operators then choose a uniform region and point the array so that the region will enter the field of view from the top and remain in the field of view as it passes underneath every pixel in the entire sensor array. Since each pixel sees the same thing, they should all produce the same intensity of light.   The satellite processes the differences between the pixel's readings and creates an adjustment factor for each pixel.   If the satellite operated in calibration mode over the Ponnequin wind farm in Colorado, then the unprocessed data would contain the information needed for this project.

THIS PAGE INTENTIONALLY LEFT BLANK

IV.   LABORATORY TECHNIQUES

A.   STATIONARY WINDMILL

1.   Imaging

The initial experiment in this lab involved a stationary windmill (the windmill is rotating, just not moving in relation to the camera) and stationary camera which was looking straight at the windmill.   This is a simplification of the overall problem which will involve a stationary windmill at Ponnequin wind farm and a moving camera (the Quickbird satellite) looking at an angle.   This simplified experiment was conducted first to work out the mathematical techniques needed to process the data.

For this experiment the webcam was set to take 300 images of the windmill as fast as it could.   The focal plane of the webcam gives images that are 240 horizontal rows and 320 vertical columns.   Eight data runs were taken and, on the first run, it took the camera 22.7 seconds to complete the 300 frames which gives a rate of 13.2 frames per second.   The webcam produced color images which were converted to grayscale to ease processing.   Note that a grayscale image is not the same as a panchromatic image from Quickbird because the grayscale image is a composite of red, green, and blue images while the panchromatic includes near infrared wavelengths.   Figure 8 shows one frame of the initial experiment.   A plain black towel was used for the background to prevent clutter in the data; however, note the white specks that appear rather dramatically in the image. These specks are simply small pieces of debris that clung to the towel.

The webcam control and data importing was done using the MATLAB mathematical assistant software created by the Math Works incorporated.   In additional to the basic software package the Image Acquisition Toolbox, and the Image Processing Toolbox were also used to facilitate data collection and processing.   The webcam is Creative Labs Webcam Plus which connects to any standard PC through a USB port.
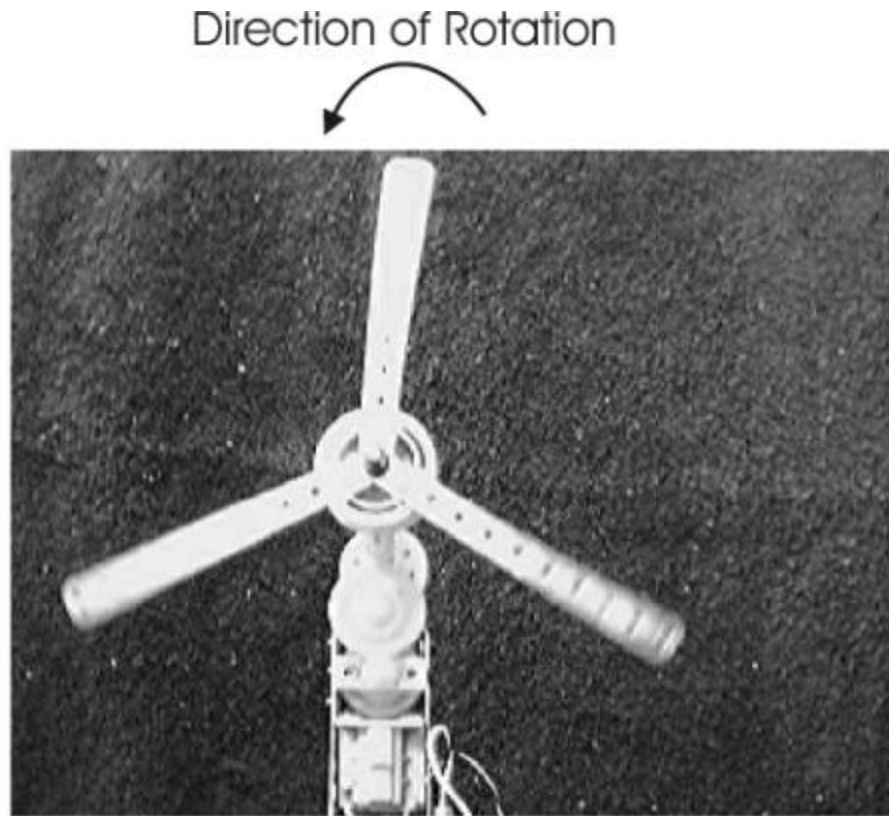
Figure 8 – Picture of model windmill during experiment.

2. Data Processing

The first step was to slice a single row out of each of the 300 frames. Any row would work, but one toward the top was chosen so the turning gears under the windmill blades would not appear in the data. This is a valid choice because the target windmills at Ponnequin Wind Farm do not have spinning gears below the blades. Figure 9 shows the row that was sliced from the data during the third iteration of the experiment. The process of extracting a single row involved two 'for' loops, an outer loop and an inner loop. The inner loop cycled 320 times which allowed it to pull each pixel of the selected row into the new array (there are 320 columns so the first iteration of the inner loop pulled the pixel located at column 1 of the selected row, the second iteration pulled the pixel at column 2, etc). The outer loop iterated 300 times and caused the inner loop to fully execute for each of the 300 frames. At this point the data is a matrix which is 320 x

300. This data represents one row through time. In the first data run it is 22.7 seconds of time. Figure 10 shows what the actual data looks like when plotted (also from the third iteration of the experiment). The row is depicted vertically on the y-axis, while the x-axis represents time. The windmill blades are bright spots in the plot. As the blade moves from right to left across the row the bright spot moves up the graph. The rotational velocity of the windmill can be determined directly from this plot. Simply count the number of lines traveling up the graph and divide by the number of blades (in this case 3), and then divide that quotient by the time.
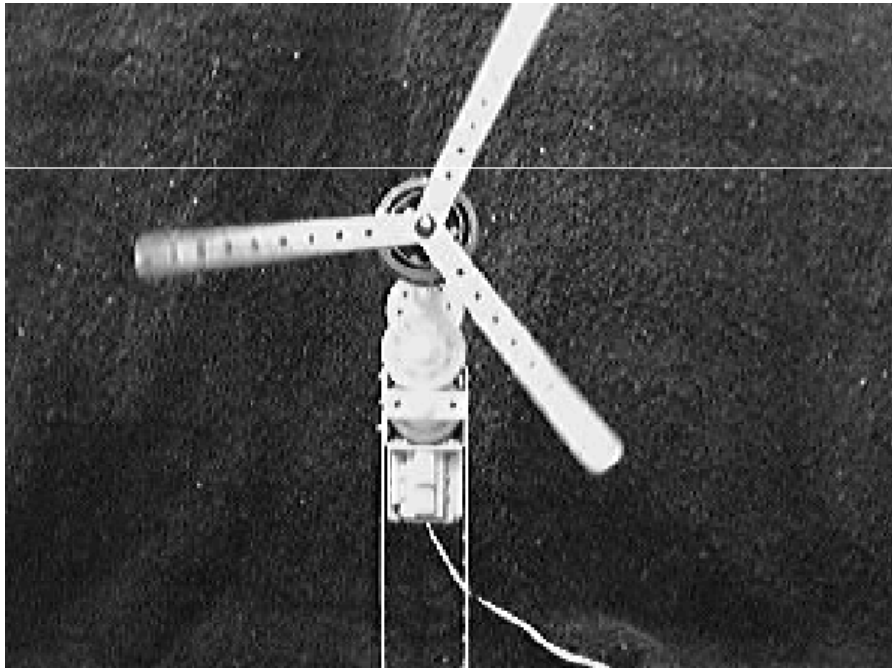
Figure 9 – Image of the windmill during the experiment. The white line represents the row that was extracted for further analysis.
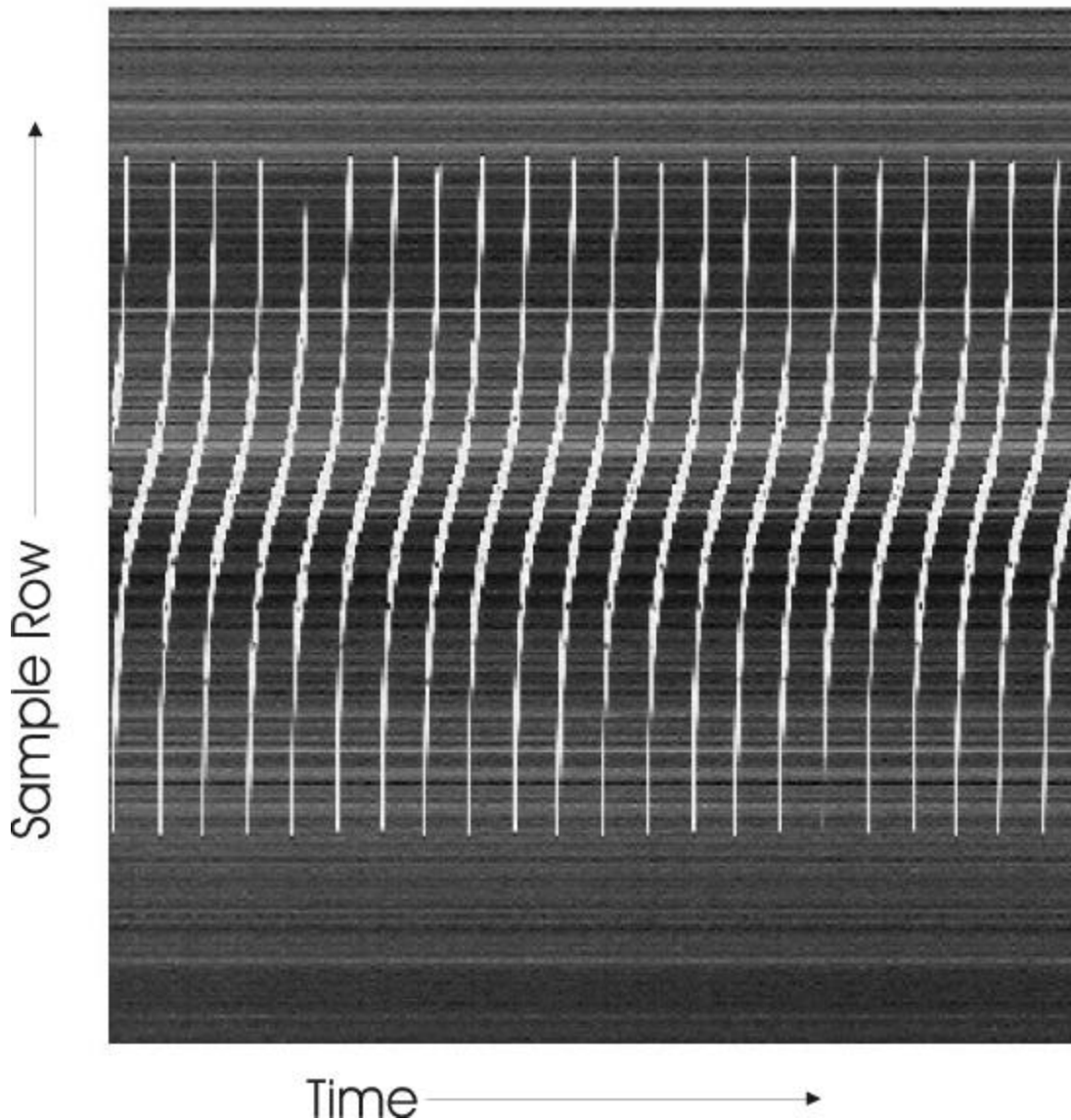
Figure 10 – This image displays the rows of data as they would come from a linear solid state array. Each vertical column in the picture is a horizontal row of data. Moving from left to right across the picture shows the rows as they occurred in time. The first row imaged is the far left, while the 300th image is the far right.

In an attempt to automate the computation of the rotational velocity the data was further processed. A single column was chosen from the original image. This column was then pulled out of the data consisting of rows over time. Think of this as a single 'for' loop executed 300 times. The first execution of the loop pulls the pixel located at

the selected column from the first frame of data, and the second iteration pulls that pixel from the second frame, etc. This leaves a matrix which is 1 x 300; it is the brightness of a single pixel over the 300 frames. Figure 11 shows the location of the chosen pixel in the original image, and figure 12 is a graph of the brightness of the pixel over time. The peaks in the graph are the times that a blade passed through the pixel. Here again, the rotational velocity can be determined by simply counting the peaks and dividing by the number of blades and the total time.

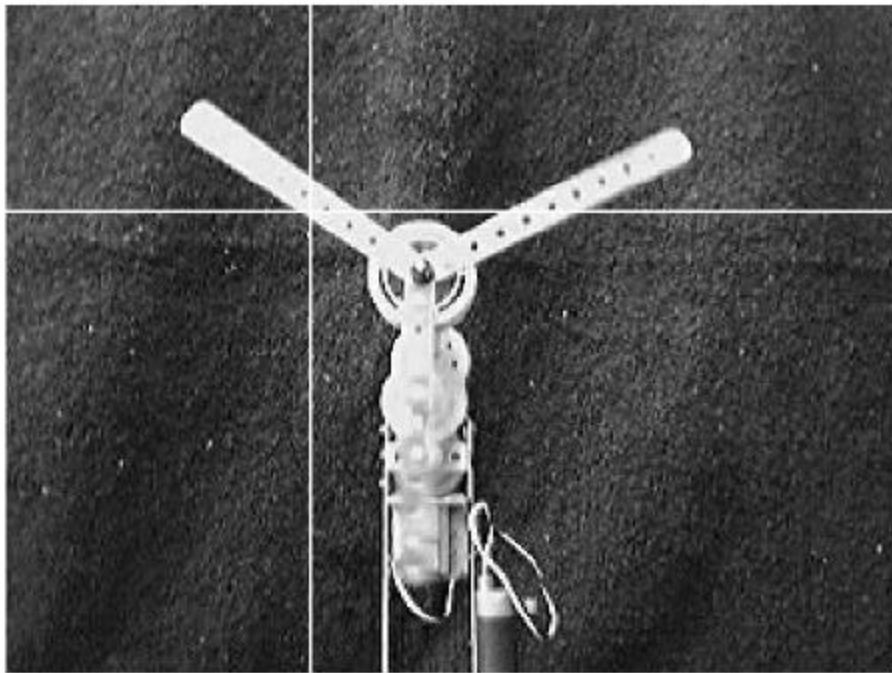**Figure 11 – This picture shows the windmill during the experiment. The horizontal white line is the row that was extracted from the data for further analysis. The vertical white line shows the column that was selected for further analysis. The pixel where the horizontal and vertical white lines meet is the only pixel carried forward to the final phase of analysis.**
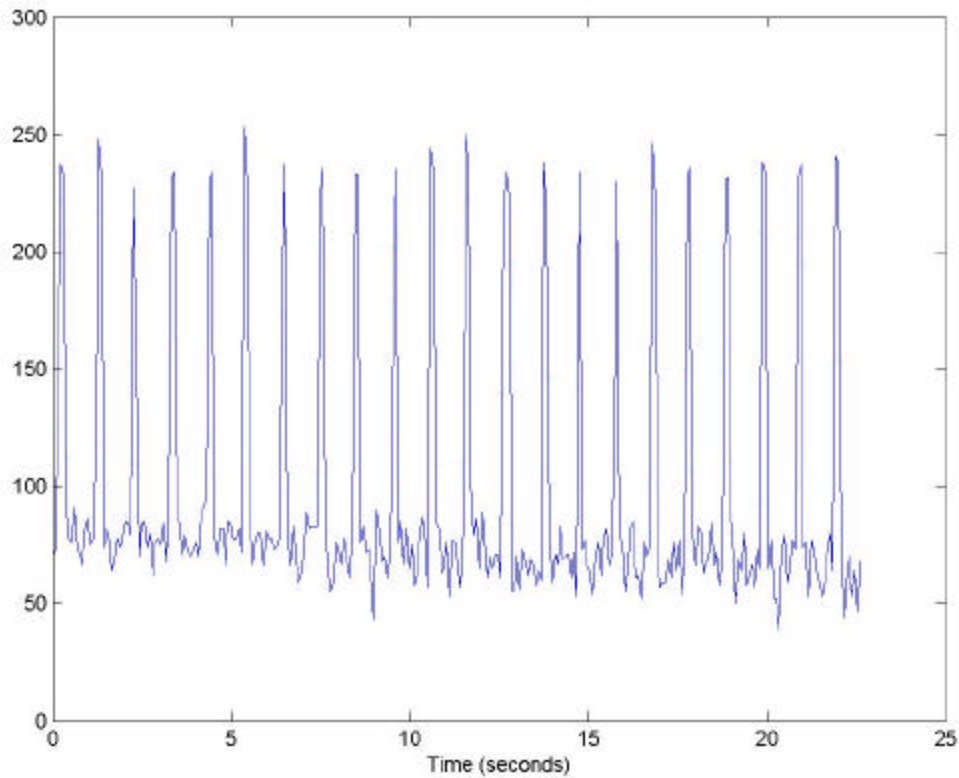
Figure 12 – Graph of pixel brightness vs. time. The y-axis is the brightness of the pixel in grayscale where 0 is pure black and 255 is pure white.

Of course, this still does not automate the process of determining the rotational velocity so a fast fourier transform of the 1 x 300 matrix was taken. The graph of the fourier transform is figure 13. Of course, taking a fast fourier transform of the data moves it from the time domain to the frequency domain. The x-axis in this graph is frequency, and the y-axis is the intensity of that frequency in the data. The large peak in the graph occurs at the origin (x = 0). The next peak occurs at X = 22 which is the number of peaks in the original graph. In other words, the second peak in the fourier transform graph occurs when X is equal to the number of times a blade on the windmill passed through the pixel under examination. The next peak in the fourier transform graph occurs when X = 44. This is double the frequency of the first peak, and it occurs because a signal with twice the frequency will produce peak values at nearly the same

22

locations as the lower frequency. The higher frequency will, of course, produce twice as many peak values as the lower frequency, but if the sampling rate of the webcam was too slow then it would not be able to measure the extra peaks. Note that the MATLAB software begins its matrices at 1 instead of 0 so when queried for which frame produced a peak in the data, MATLAB will return an integer that is the frequency plus 1. This is, of course, easily corrected by simply subtracting 1 from the result returned by MATLAB.
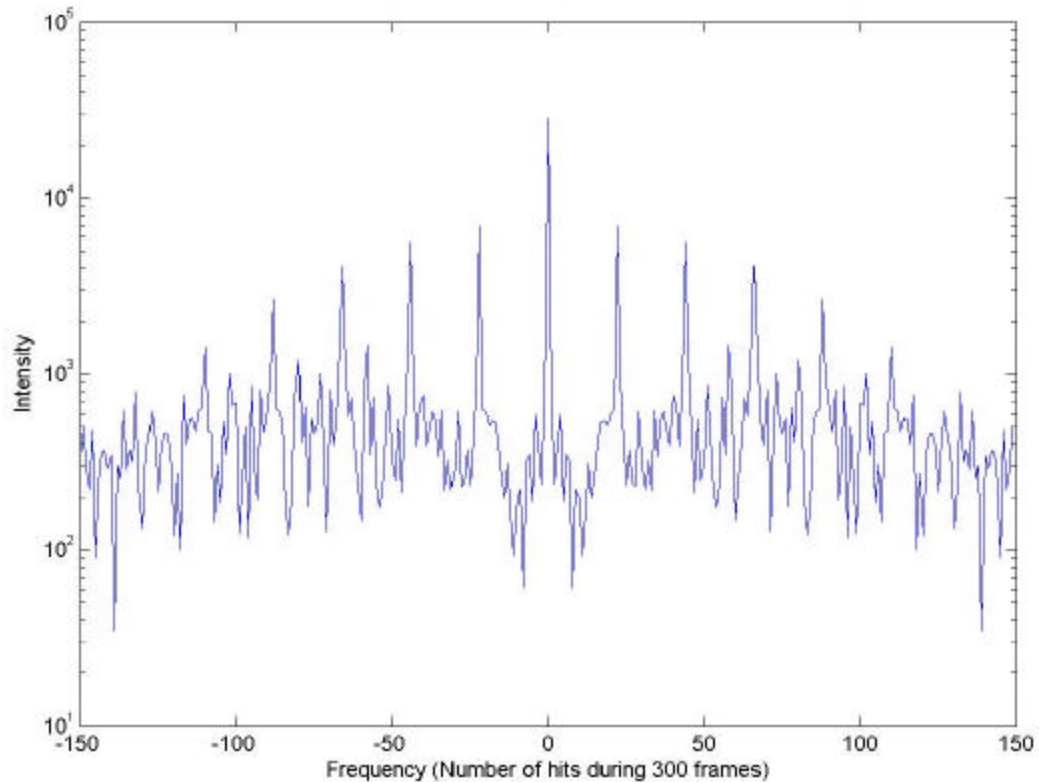
Figure 13 – Fourier Transform of the brightness of one pixel through 300 frames of data. The X-axis is frequency, and the Y-axis is the intensity of the frequency in the data. Note that the Y-axis is a log scale.

3.   Results

The actual rotational velocity of the windmill varied from one data run to the next because the windmill is powered by two 9 volt batteries. As the charge on the batteries depleted, the rotational velocity of the windmill decreased. This made it necessary to

measure the rotational velocity of the windmill with a stopwatch for each of the eight data runs. The data from all eight runs is summarized in table 1. The Total Time column displays the amount of time that MATLAB used to acquire all 300 frames of data for each iteration of the experiment. Note that the total time varies slightly from one iteration to the next; this is caused by delays either in the webcam (possibly due to thermal variations) or inside the computer. Delays inside the computer could be caused by different memory allocations or thermal variations in the CPU or RAM memory. The variations in total time could not be controlled with this importing technique, but they could be measured so they have a minimal effect the accuracy of the data.

The next column, Hits, is a count of how many times a blade of the windmill passed through the pixel being examined. It was obtained from the fourier transform of the pixel brightness over time array. Dividing Hits by 3 (because there are 3 blades on the windmill) gives the number of rotations of the windmill during the experiment. This causes a source of error that can not be corrected with this image acquisition and processing technique. The error is the result of having an integer number of blade hits. For example, in the first data run, the windmill may have only conducted 7.45 rotations instead of 7.67, but because the position of the windmill blades when the imaging began, 23 hits were recorded. This would cause the measured rotations per second to be too high. The effect of this error is small in this experiment due to the long duration of each data run (over 20 seconds each). If each data run lasted only 5 seconds, then the windmill would complete less than 2 rotations so the effect of being off by 0 - .33 rotations would have a much greater effect.

The measured rotations per second data was obtained by dividing the number of rotations by the total time. The actual rotations per second data was obtained by using a stopwatch to time how long it took for the windmill to complete 30 rotations and then dividing. Note that the 30 seconds used to calculate the actual rps is longer than the total time used by MATLAB to acquire its data. In each case, the windmill was started (or left running for iterations 2-8), and then the execution command was entered into MATLAB. After MATLAB began, the stopwatch was started for computing the actual rps.

24

The formula used to compute the percent difference is:

$$\%Difference = \frac{Actual\_rps - Measured\_rps}{Actual\_rps}$$

Data Run	Total Time	Hits	Rotations	Measured rps	Actual rps	% Difference
1	22.75 s	23	7.67	0.3370	0.3293	2.34%
2	22.72 s	22	7.33	0.3228	0.3248	0.61%
3	22.68 s	22	7.33	0.3233	0.3196	1.15%
4	22.59 s	25	8.33	0.3689	0.3664	0.67%
5	22.56 s	25	8.33	0.3694	0.3672	0.59%
6	22.49 s	25	8.33	0.3705	0.3670	0.96%
7	22.52 s	25	8.33	0.3700	0.3672	0.76%
8	21.47 s	23	7.67	0.3571	0.3587	0.44%

Table 1 – Summary of Data from 8 iterations using a stationary windmill.

B. MOVING WINDMILL

1. Imaging

Of course, the satellite will be moving when it images the Ponnequin Wind Farm, so the next step in the laboratory simulations required simulating the satellite movement. Fortunately, for the laboratory experiment, moving the webcam at a constant velocity during the imaging process produces the same data as moving the windmill at constant velocity. It was quickly determined that moving the windmill would be easier because the Erector sets from which it was built came with wheels and axels which could easily be attached. Moving the webcam would have been more difficult because the USB cable connecting it to the computer was not very long so it could only collect images over a

short distance. The mobile windmill along with a small car which tows it is shown in figure 14.

Figure 14 – Mobile windmill and towing vehicle

The moving windmill data runs were still done by setting the webcam to take 300 images. The webcam was activated while the windmill was outside its field of view (off to the right), and the windmill passed completely through the webcam's field of view before the 300 frames were complete. This means there are frames at the beginning and end of the data that do not have the windmill in them at all. Moving the windmill across the field of view of the webcam from right to left forced the windmill to slide across the rows of the webcam's rectangular solid state array. This means that the horizontal rows of the rectangular array see the target in the same way that Quickbird's linear sensor array will in calibration mode. Data runs were taken with the windmill blades spinning clockwise and counterclockwise so the data could be compared.

2. Data Processing

The data processing for the moving windmill is very similar to that for the stationary windmill. First, the 300 frames of color images were converted into 300 frames of grayscale images. Next, a single row was removed from each of the 300 images and compiled into a 320 x 300 matrix. Again, the row that was extracted was above the center pivot of the windmill. Like before, this is a laboratory representation of the actual data that the satellite will return. Figure 15 shows the data when the windmill was spinning counterclockwise. Note that bright lines still appear in the image, but they are not as bright as the stationary data, and they seem to travel up the image. They are less bright because the instantaneous velocity of any point on the arm is faster than it was with the stationary windmill. This increase in velocity occurs because the lateral component of the arm's movement is in the same direction as the windmill when the arm is above the center pivot. The phenomenon just described is often referred to as a Doppler shift.
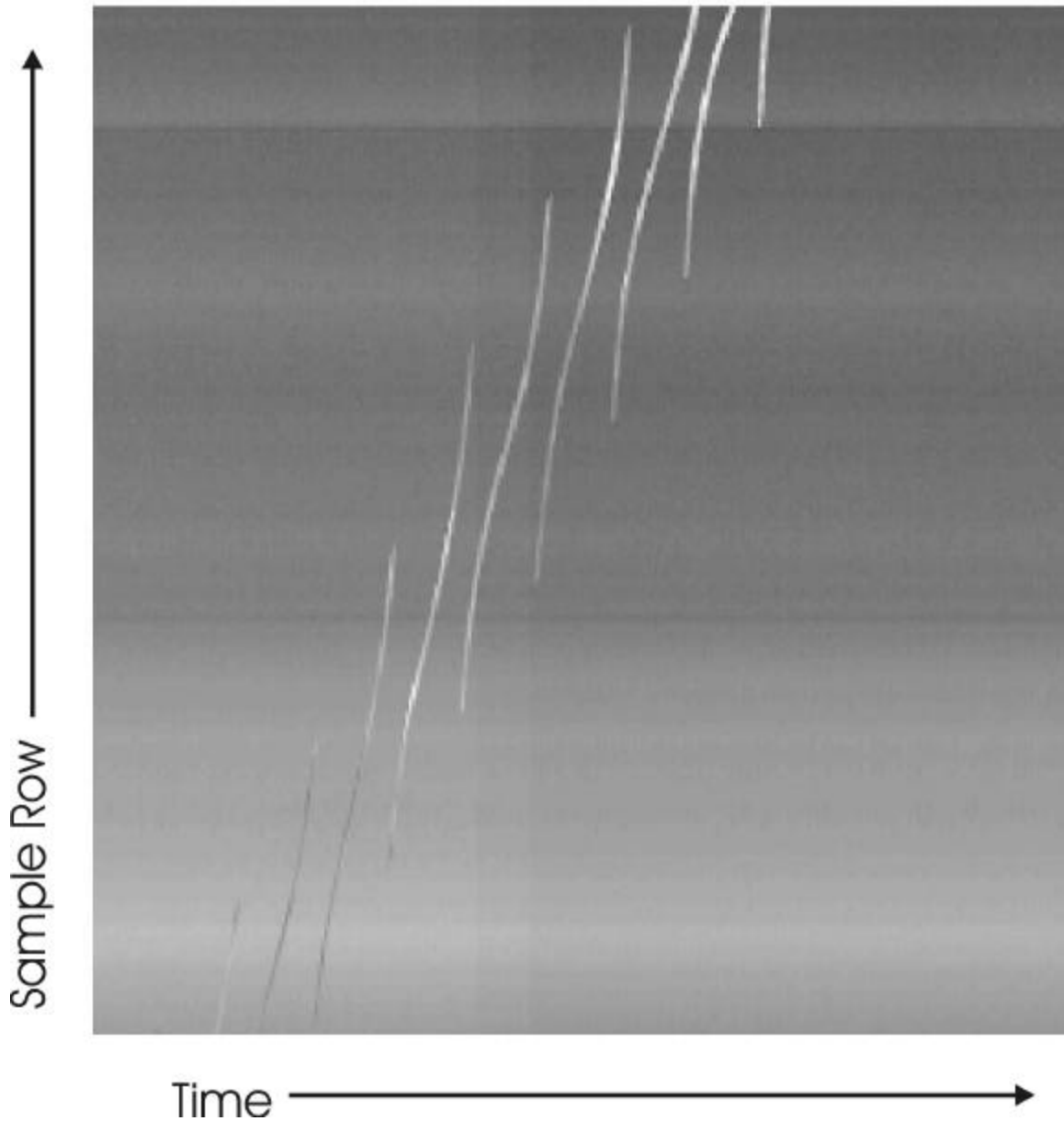
Figure 15 – Simulated satellite data with a moving windmill spinning counterclockwise. This picture consists of 300 frames of a single row of data from the rectangular array (webcam). Each vertical column is a row of data. The first image taken is the column on the far left. The 300th image taken is the row on the far right.

Figure 16 shows the data with the windmill still moving from right to left, but now the arms are spinning in a clockwise direction. This time the bright spots are short and 'S' shaped. This occurred because the instantaneous velocity from rotation is opposite in direction to the velocity from the movement of the entire windmill. Of

28

course, the instantaneous horizontal velocity on a point of the arm from rotation varies as the arm moves around the circle. For example, when the arm is pointing straight up, the velocity of the tip of the arm is completely to the right (there is no vertical velocity). After that arm rotates 90° and is pointing straight to the right, the entire velocity of the tip is going down (there is no horizontal velocity). At all points in between those two extremes, the velocity of the tip has components in the horizontal and vertical directions.
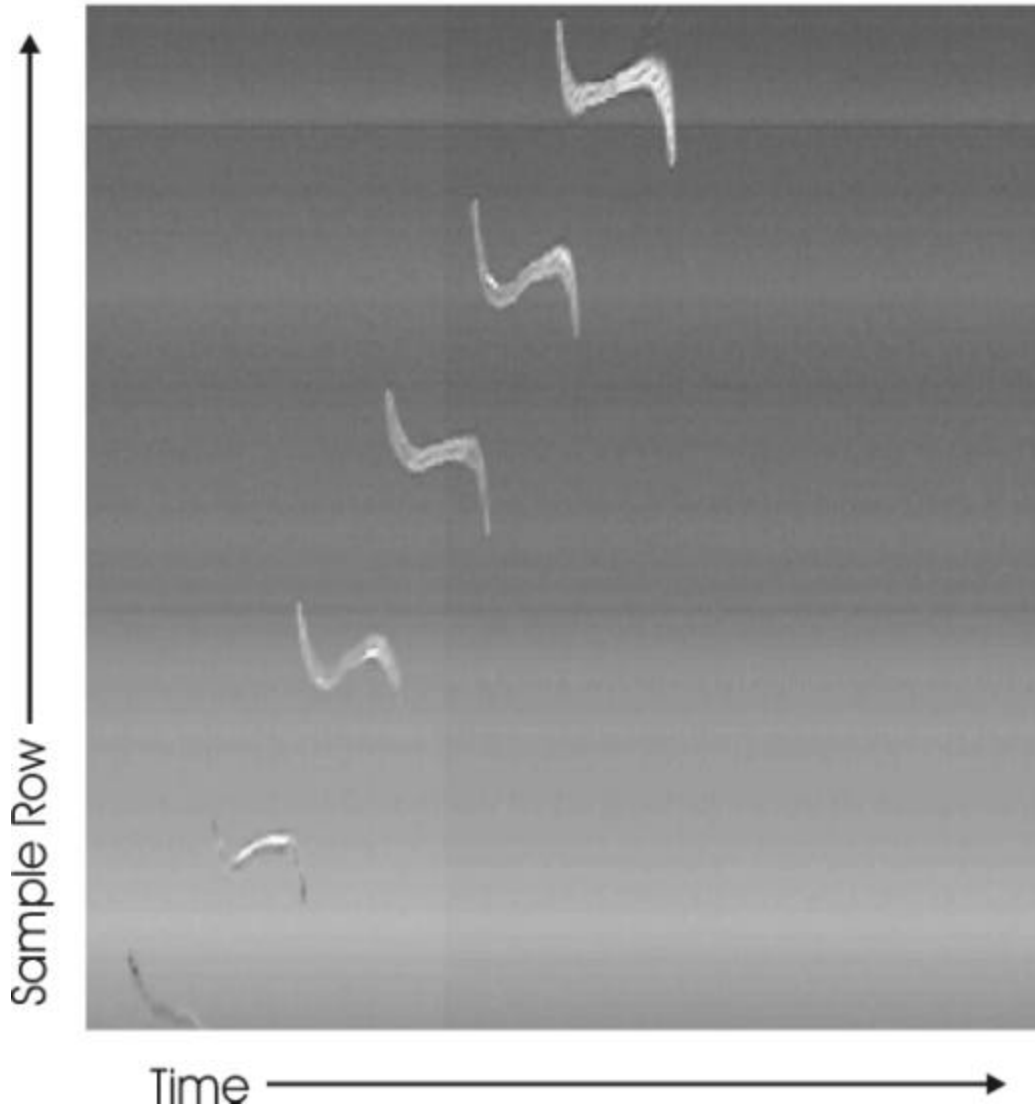

Figure 16 – Simulated satellite data with a moving windmill spinning clockwise. This picture consists of 300 frames of a single row of data from the rectangular array (webcam). Each vertical column is a row of data. The first image taken is the column on the far left. The 300th image taken is the row on the far right.

The data in figure 16 has an 'S' shape because when the arm moves into the row that is being extracted its rotational velocity is going to the right (and up, but the vertical component is not important here) while the linear motion of the windmill is going left. The horizontal component of the rotational velocity is greater than the linear velocity which makes the part of the arm being imaged appear to move to the right when viewed from the stationary camera. This makes the bright line move downward initially in figure 12 (the bottom of the picture is the far right of the camera's field of view, and the top is the far left. Another factor influencing the apparent motion of the arm is the fact that as the windmill rotates, the part of the arm that is crossing the extracted row moves toward the center pivot. When an arm is pointed straight up the point on the arm being imaged is closest to the center pivot of the windmill. As the windmill continues to rotate the point on the arm that is being imaged moves away from the center to the tip. This causes the bright spot in the row to move to the right even faster as the arm sweeps downward which causes the downward arc of the 'S' shapes.

3.   Results

A total of six iterations were conducted with the windmill moving. The blades of the windmill spun clockwise for the first three iterations, and they spun counterclockwise for the last three. The data from the six runs is contained in table 2.

Data Run	Total Time (sec)	Hits	Rotations	Measured rps	Actual rps	% Difference
1	4.4476	6	2	0.4497	0.4401	2.19 %
2	4.2233	6	2	0.4736	0.4659	1.65 %
3	4.1585	6	2	0.4809	0.4773	0.76 %
4	2.2072	3	1	0.4531	0.4724	4.09 %
5	2.144	3	1	0.4531	0.4723	1.24 %
6	2.688	4	1.33	0.4960	0.4766	4.08 %

Table 2 Summary of data from moving windmill experiments. The first three data runs have a clockwise rotating windmill, and the last three have a counterclockwise rotation.

There is an important difference in the process for calculating the measured rotations per second with the moving windmill than with the stationary one.   The difference comes from the fact that there is idle time at the start and end of the imaging process where the windmill is not in the field of view of the webcam.   These idle frames had to be trimmed from the data to determine the total time that the windmill was actually imaged (see figure 17).   MATLAB keeps an exact time on every frame of data that it acquires from the webcam so the frame number where the windmill enters the field of view and the frame number where the windmill exits the field of view must be determined.   After those two numbers are found the total time of image acquisition is simply the time for the exit frame minus the time of the entrance frame.
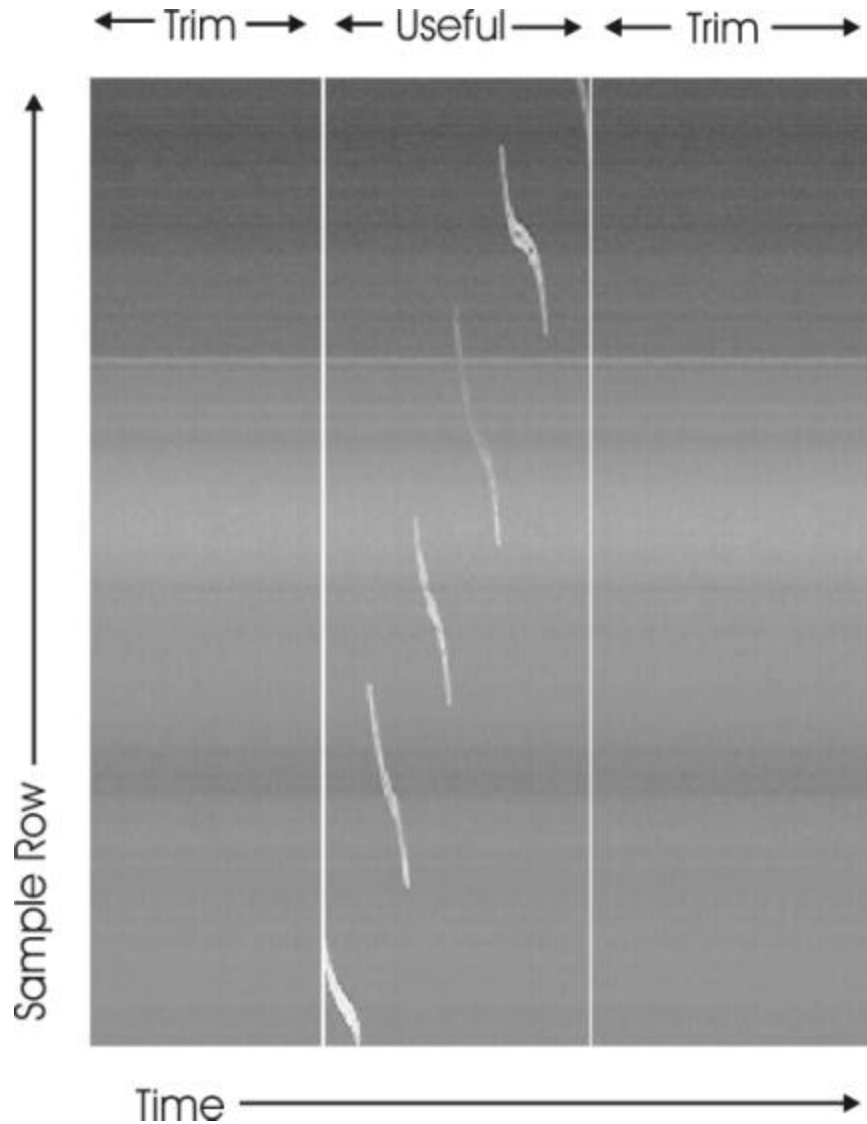
Figure 17 – Trimming useful data from iteration 3 where the windmill is spinning clockwise. The vertical line on the left is the frame where the windmill enters the field of view of the webcam. The vertical line on the right is the frame where the windmill leaves the field of view. The frames of data between the two lines hold the data which was processed to produce the data in table 2.

Unfortunately, there are not enough hits contained in the data to make a two dimensional fourier transform useful. In the above picture there are only six hits which is so close to zero that the frequency response looks like noise. For this reason, the number of hits listed in table 2 comes from manually counting the number of hits in the image. Speeding up the windmill's rotational velocity could have produced enough hits to make a fourier transform useful, but the frame rate of the webcam used in the experiment is too

32

slow to work with a faster windmill. Number of rotations, measured rps, and actual rps, and percent difference are all calculated in the same way as the stationary windmill data.
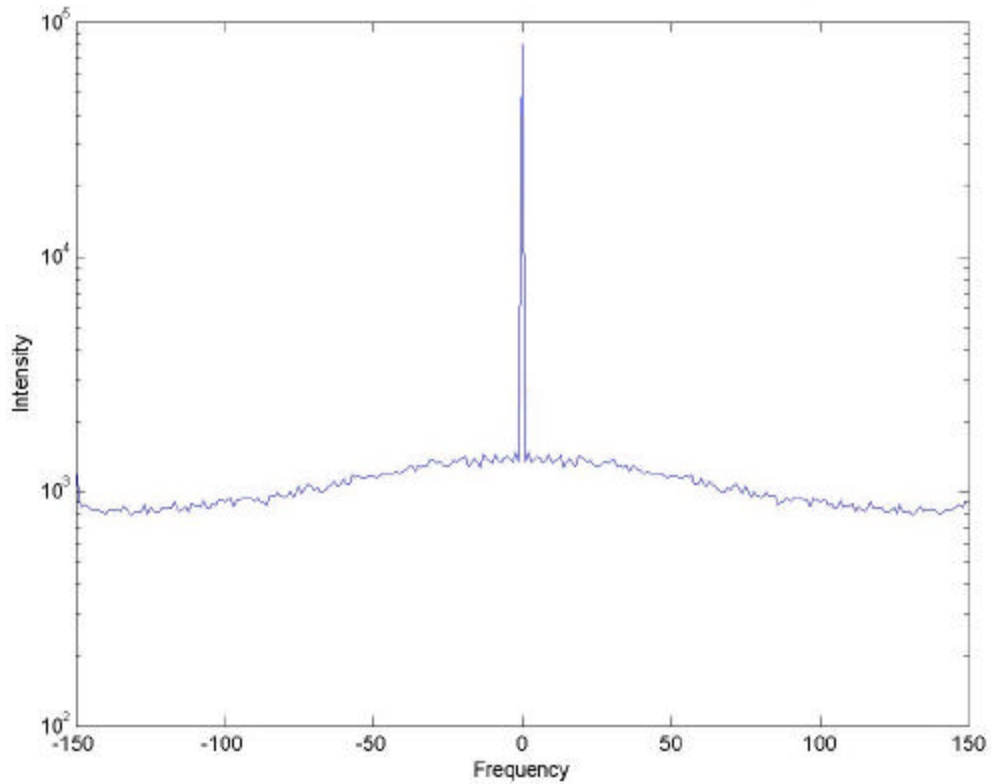

Figure 18 – Fourier transform of 300 frames of data from a moving windmill with blades spinning clockwise. Note that noise dominates any frequency responses that may be present in the data due to the low number of blade hits. This data was obtained by conducting a fourier transform on each frame of data and then averaging all 300 frames together.
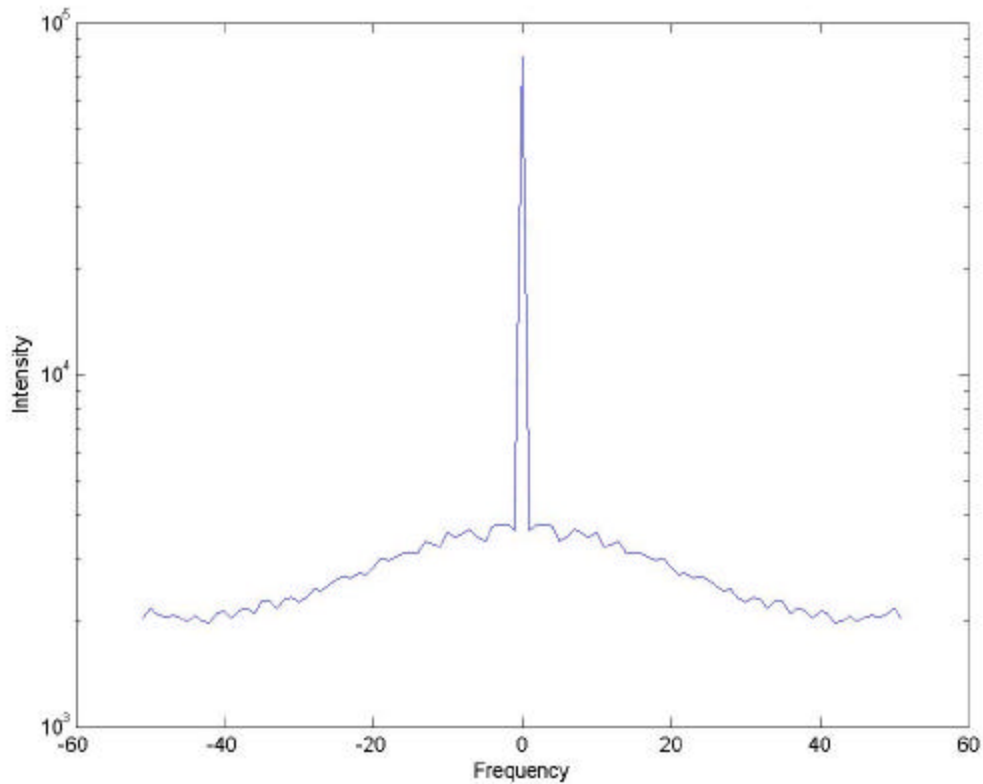
Figure 19 – Fourier transform of data from a moving windmill with idle frames trimmed from the beginning and end of the data. There are a total of 103 frames of useful data in the image. Note that here again, noise dominates over any frequency responses that may be present in the data. This data was obtained using the same averaging technique described in figure 18.

The counterclockwise data required a little more work in the trimming process. The hits in the counterclockwise iterations are much longer and only a couple of the hits completely fall inside the imaged row. This is best understood by looking at figure 20 where there are a total of eight blade hits in the image. Unfortunately, only two of those hits fall completely within the image itself. The complete hits represent the most useful data because each of them truly represents one third of a rotation of the windmill. One partial hit in front of the full hits and one partial hit behind them can still be used as full hits, but any hits beyond them must be discarded because they are caused by the Doppler shift from the moving windmill instead of the actual rotation of the blades.
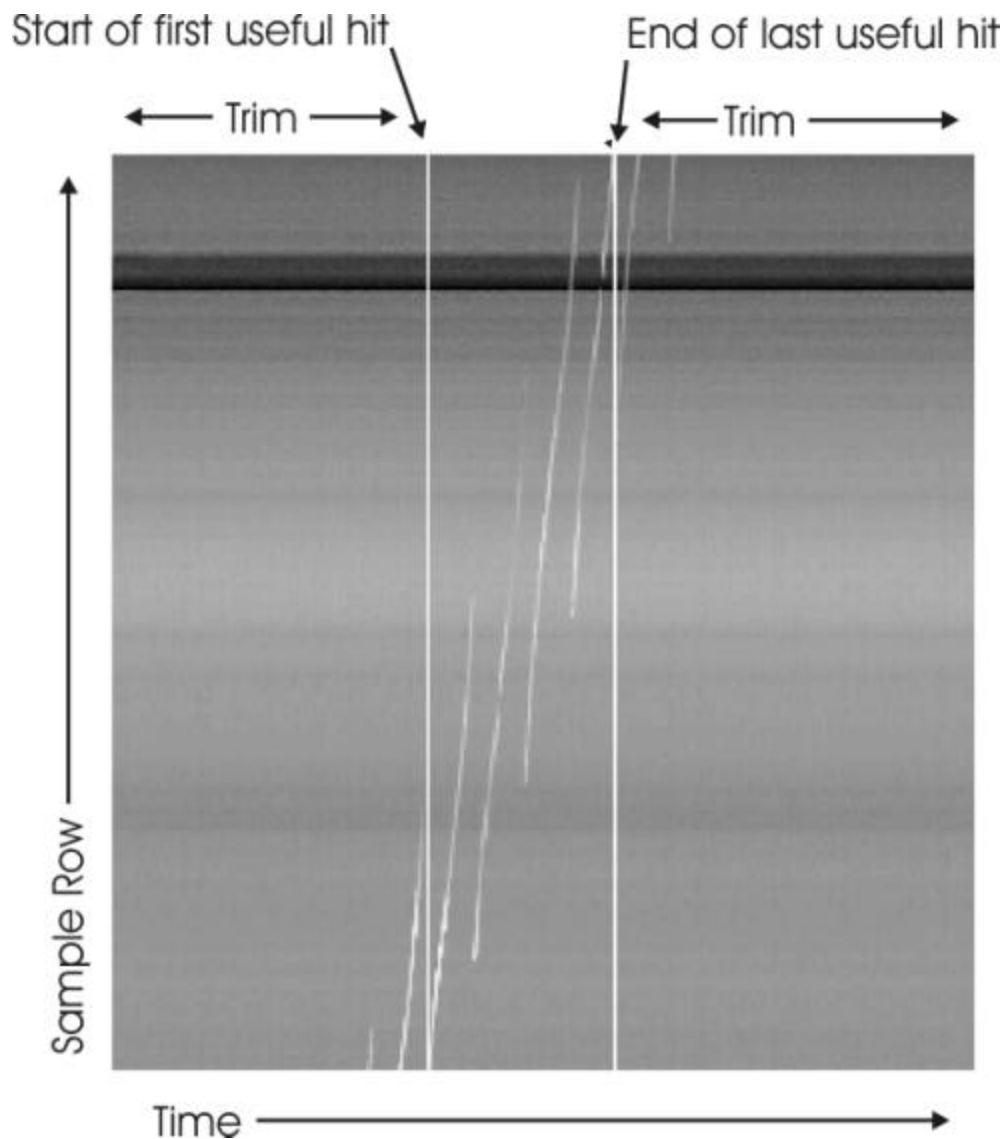
Figure 20 – This figure depicts trimming the useful data from iteration 6 where the windmill is moving and its blades are spinning counterclockwise. The first two and last two partial hits had to be discarded because they are caused by the Doppler shift from the moving windmill, not from the rotational velocity of the blades themselves.

THIS PAGE INTENTIONALLY LEFT BLANK

V.   APPLYING TECHNIQUE TO SATELLITE DATA

A.   PONNEQUIN WIND FARM

Ponnequin wind farm is located in northeastern Colorado just south of Cheyenne, Wyoming.   It is run by Excel Energies and is capable of producing 30 megawatts of electricity from 44 wind turbines.   There are two types of windmill on the farm: Veritas and Micon.   Dimensionally, both types of windmill are about the same, standing 181 feet high (to the center pivot of the windmill).   The windmill blades have a diameter of 159 feet.   The Veritas windmills operate around 30 rotations per minute while the Micons generally run 22 rpm.   Figure 21 shows some of the windmills at sunset, and figure 22 shows a row of windmills during the day.

Figure 21 – Ponnequin Wind Farm at sunset

Figure 22 – Ponnequin Wind Farm during the day

The windmills are arranged in 4 lines which are oriented in a generally southeast fashion. The ground on the farm is relatively level, and the spacing between windmills is fairly consistent. The windmills turn into the wind so predicting their exact orientation is impossible, but a good estimate can be found by watching the weather forecast for the area. Maintenance personnel are on site almost every weekday, and they have access to a computer that monitors the rotational velocity of each windmill. The velocity and electrical output of each windmill is not recorded; however, so the actual rotational velocities of the windmills will have to be recorded by someone at the windfarm during the time of data acquisition. Figure 23 shows the actual layout of the windmills.
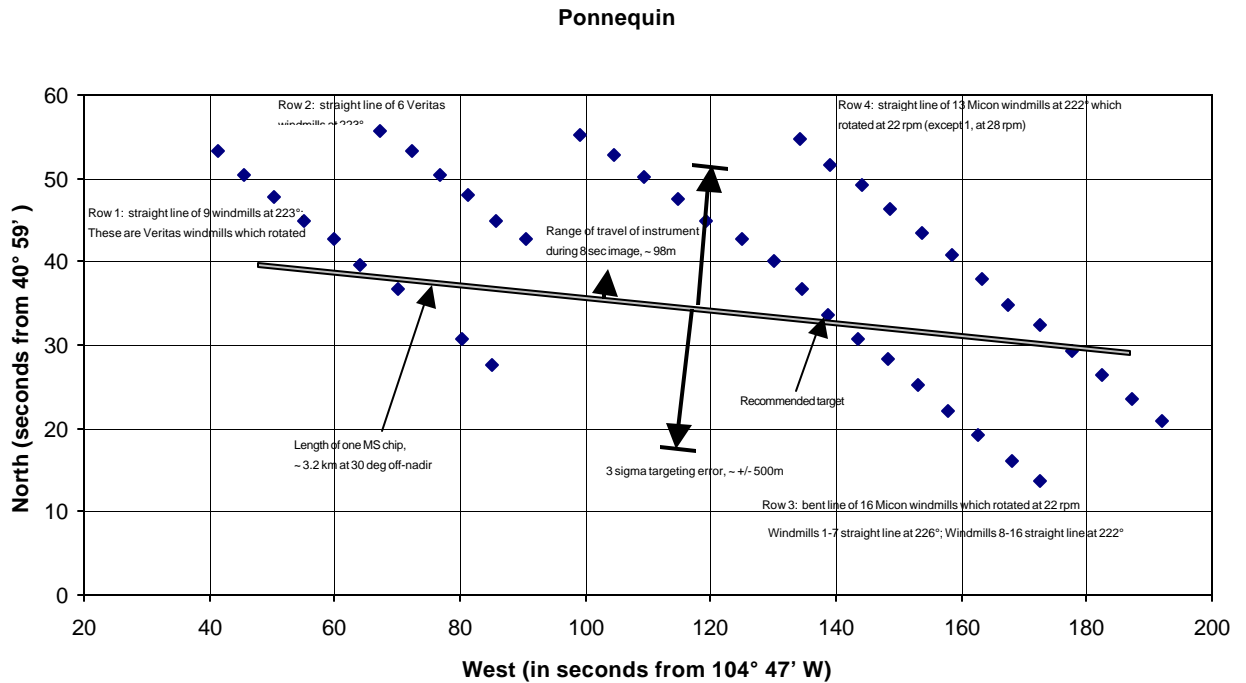
Ponnequin

Figure 23 – Ponnequin Wind Farm layout

B.   SATELLITE SPECIFICS

The satellite is not capable of looking directly at the windmills like the webcam did.   Instead, it will have a look angle of about 30° off nadir.   This should still give enough of an angle to make the motion of the blades visible in the data.   The long gray line in figure 23 represents the distance on the ground that one chip from the multispectral array sees when imaging 30° off nadir.   The error bars going up and down from it represent the pointing error of the satellite.   This means several different windmills should appear in the data; however, it will be difficult (if not impossible) to determine exactly which windmills the satellite captured.   This means the rotational velocities of all windmills at the time of data acquisition should be recorded.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

This thesis demonstrates the feasibility of using a commercial, high resolution imaging satellite to detect and measure temporal phenomenon on the surface of the earth. Specifically, two different signatures representing rotational movement were isolated, and their basic movement characteristics were explained in simulated satellite data. Rotational velocities were extracted from the simulated data, and in every data run, this extracted rotational velocity was within 5% of the actual rotational velocity. It is important to note that the techniques used in this thesis were only intended to demonstrate the feasibility of the overall project. Follow on research involving actual satellite data will be processed in a similar manner, but the exact technique used will, of course, have to be modified as unforeseen elements show up in the satellite data.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Holst, Gerald C., <u>CCD Arrays Cameras and Displays</u>,. p 45, JCD Publishing, Winter Park, FL, 1998.

[2] Holst p 46

[3] Holst p 46

[4] Holst p 48

[5] Holst p 48

[6] Holst p 49

[7] Holst p 49

[8] Holst p 70

[9] Holst p 71

[10] Olsen, Richard C., <u>Remote Sensing from Air and Space</u> p 81, Sep 2002.

[11] Olsen p 82

[12] Digital Globe Incorporated, 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Dudley Knox Library
    Naval Postgraduate School
    Monterey, California

2. Professor Richard Olsen
    Naval Pograduate School
    Monterey, California

3. Professor Richard Harkins
    Naval Pograduate School
    Monterey, California

4. Andrew Hittner
    Naval Pograduate School
    Monterey, California